

16623 - Advanced Computer Vision Apps

Assignment 3 - Lucas & Kanade Algorithm, Correlation Filters - (15 % of total grade) 100 points - (Q1, Q2, Q3)

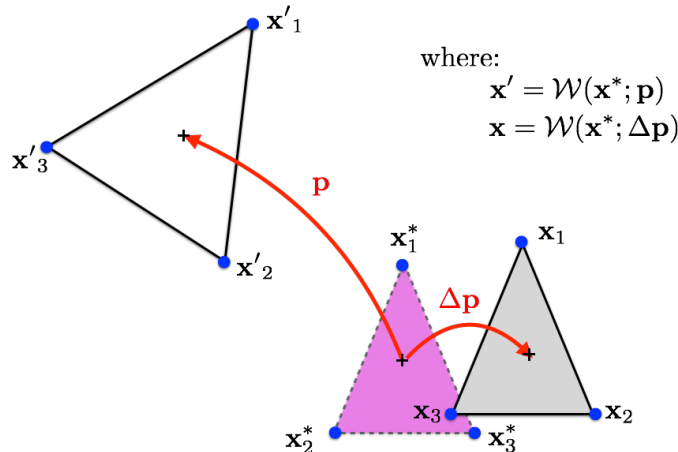
Released - 19th of October

Due - Wednesday 2nd of November (midnight EST)

On your local machine create a directory called `Assignment_3`. Once you have completed the assignment questions below, zip up the contents of the `Assignment_3` directory and upload them to your AFS drop-box `/afs/cs.cmu.edu/academic/class/16623-f16-users/andrew.id/assignment3`. For any portions of these questions that require a written response please create a `WrittenResponses.pdf` document in the `Assignment_3` directory before zipping up the contents (please label clearly inside the document which question you are answering).

1) Download the MATLAB project Lucas-Kanade (see link <https://github.com/slucy-cs-cmu-edu/Lucas-Kanade>). Run the script file `example.m`, and observe the LK algorithm in action. Inspect Lectures 12 & 13 and see if you can follow all the steps in the code. In this assignment you will be mainly modifying the `LK.m` and `LK_IC.m` object classes. Please proceed to answer the following questions:-

- a) In the `LK.m` object class change the Gaussian blur variance in `fspecial('gaussian',[5,5],3)` from 3 to 0.1. Save the changes, and then re-execute the `example.m` script file. What do you notice about the performance of the algorithm? Why do you think the blur effects the performance of the algorithm? Finally, what would be the advantage (from a computational standpoint - think SIMD here) of replacing the Gaussian filter with a box filter using `fspecial('average',[5,5])`? Capture the result of this graphical output after employing the box filter. **Include all answers and images in your written response. (5 points)**
- b) Inspect the figure below. Assume the $N = 3$ points $\{\mathbf{x}'\}_{n=1}^N$ stem from the source image coordinates,



and the points $\{\mathbf{x}\}_{n=1}^N$ stem from the template coordinates. The above figure depicts how these two coordinate frames are indirectly related through the proxy coordinates $\{\mathbf{x}^*\}_{n=1}^N$. Using the knowledge that for an affine warp,

$$\mathbf{x}' = \mathcal{W}(\mathbf{x}^*; \mathbf{p}) \rightarrow \tilde{\mathbf{x}}' = \mathbf{M}\tilde{\mathbf{x}}^*$$

$$\mathbf{x} = \mathcal{W}(\mathbf{x}^*; \Delta \mathbf{p}) \rightarrow \tilde{\mathbf{x}} = \Delta \mathbf{M}\tilde{\mathbf{x}}^*$$

where $\tilde{\mathbf{x}} = [\mathbf{x}^T, 1]^T$ is the representation of $\mathbf{x}$ in homogeneous coordinates. Demonstrate with full mathematical working how one would update $\mathbf{M}$ using $\Delta\mathbf{M}$ such that $\tilde{\mathbf{x}}' = \mathbf{M}\tilde{\mathbf{x}}$? **Show the mathematical working and all answers in your written response. (5 points)**

- c) The result to the previous question is important as it demonstrates how we can determine the direct relationship between the template image $\mathcal{T}(\mathbf{0})$ and the source image $\mathcal{I}$ when we obtain the solution to the indirect relationship,

$$\arg \min_{\Delta\mathbf{p}} \|\mathcal{I}(\mathbf{p}) - \mathcal{T}(\Delta\mathbf{p})\|_2^2$$

which is approximately linearized in practice as,

$$\arg \min_{\Delta\mathbf{p}} \|\mathcal{I}(\mathbf{p}) - \mathcal{T}(\mathbf{0}) - \frac{\mathcal{T}(\mathbf{0})}{\mathbf{p}^T} \Delta\mathbf{p}\|_2^2 .$$

Based on this result, the Lecture 12-13 notes and reading materials fill in the missing code within the `LK_IC.m` object class file so as to implement an inverse compositional update. Use the class functions `p2M` and `M2p` in the provided `Affine.m` object class to help you with this. If everything is working properly you should be able to swap the `LK` object for the `LK_IC` object in the `example.m` script file and re-run everything. Capture the result of this graphical output, this time displaying the bounding box in green. In your own words please describe why the inverse compositional approach is more computationally efficient than the classical forwards additive approach? **Include all answers and images in your written response. (20 points)**

- d) In the `LK_IC.m` object class one can note that the `imfilter` operation is being used with the “replicate” flag being set. Turn the flag off (i.e. just remove it from the function’s input). What does this do to the result? Why is this not an issue with the forwards additive method in the `LK.m` object class? What could be done to improve the performance of the inverse composition method further in this regard (think of boundary effects and possibly the employment of the `conv2` function with the “valid” flag)? **Include all answers in your written response. (5 points)**

- 2) Download the MATLAB project `Corr-Filters` (see link <https://github.com/slucy-cs-cmu-edu/Corr-Filters>). Run the script file `example.m`, and observe a visual depiction of the extraction of sub-images from the Lena image (Figure 1 in MATLAB) as well as the desired output response from our linear discriminant (Figure 2 in MATLAB). Inspecting `example.m` you can see that it extracts a set of sub-images $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^T$ from within the Lena image. These sub-images are stored in vector form so that $\mathbf{x}_n \in \mathbb{R}^D$ (in the example code all the sub-images are 29×45 therefore $D = 1305$). Associated with these images are desired output labels $\mathbf{y} = [y_1, \dots, y_N]$ where y_n lies between zero and one. For this example we have made $D = N$ on purpose.

Please proceed to answer the following questions:-

- a) A linear least-squares discriminant can be estimated by solving

$$\arg \min_{\mathbf{g}} \sum_{n=1}^N \frac{1}{2} \|y_n - \mathbf{x}_n^T \mathbf{g}\|_2^2$$

we can simplify this objective in vector form and include an additional penalty term

$$\arg \min_{\mathbf{g}} \frac{1}{2} \|\mathbf{y} - \mathbf{X}^T \mathbf{g}\|_2^2 + \frac{\lambda}{2} \|\mathbf{g}\|_2^2 . \quad (1)$$

Please write down the solution to Equation 1 in terms of the matrices $\mathbf{S} = \mathbf{X}\mathbf{X}^T$, $\mathbf{X}$ and $\mathbf{y}$. **Place the answer in your written response. (5 points)**

- b) Add your solution to Equation 1 in the `example.m` code. Visualize the resultant linear discriminant weight vector $\mathbf{g}$ using the MATLAB function `imagesc` for the penalty values $\lambda = 0$ and $\lambda = 1$ (remember to use the `reshape` command to convert $\mathbf{g}$ back into a 2D array). Apply the filter to the entire Lena image using the `imfilter` function. Visualize the responses using `imagesc` for both values of λ . Include these visualizations in your written response document. Can you comment on which value of λ performs best and why? **Place your answers and figures in your written response. (5 points)**
- c) Visualize the response you get if you attempt to use the 2D convolution function `conv2` with the “same” flag. Why does this get a different response to the one obtained using `imfilter`? How could you use the MATLAB operations `flipud` and `fliplr` to get a response more similar to the one obtained using `imfilter`? **Place the answer in your written response. (5 points)**

3) Inspect the following properties of circulant Toeplitz matrices and the Fourier transform.

Property 1: the $D \times D$ Fourier transform matrix $\mathbf{F}$ is a scaled orthobasis therefore

$$\|\mathbf{x}\|_2^2 = \frac{1}{D} \|\hat{\mathbf{x}}\|_2^2$$

which is commonly referred to as Parseval’s theorem. Since $\hat{\mathbf{x}} = \mathcal{F}\{\mathbf{x}\} = \mathbf{F}\mathbf{x}$ is the Fourier transform therefore $\mathbf{x} = \mathcal{F}^{-1}\{\hat{\mathbf{x}}\} = \frac{1}{D}\mathbf{F}^T\hat{\mathbf{x}}$ is the inverse Fourier transform.

Property 2: that a circulant Toeplitz matrix $\mathbf{X}$ can be formed from any vector $\mathbf{x} \in \mathbb{R}^D$ such that,

$$\mathbf{X}(m, n) = \mathbf{x}(\text{mod}\{m - n, D\}) \text{ .}$$

See a description of the `mod` operator/function in MATLAB.

Property 3: that a circulant Toeplitz matrix can always be diagonalized by,

$$\mathbf{X} = \frac{1}{D}\mathbf{F}^T \text{diag}(\hat{\mathbf{x}})\mathbf{F} = \frac{1}{D}\mathbf{F} \text{diag}(\text{conj}\{\hat{\mathbf{x}}\})\mathbf{F}^T$$

where $\mathbf{x} \in \mathbb{R}^D$ is the vector from which the circulant Toeplitz matrix $\mathbf{X}$ was formed.

Property 4: that the multiplication of any two circulant Toeplitz matrices $\mathbf{X}$ and $\mathbf{Y}$ is itself a circulant Toeplitz matrix $\mathbf{Z}$,

$$\mathbf{X}\mathbf{Y} = \mathbf{Z} \text{ .}$$

Property 5: properties 1-4 apply not only to circular shifts in one dimension, but to N -dimensional circular shifts. For example a circulant Toeplitz matrix $\mathbf{X}_{2D}$ formed from 2D circular shifts is diagonalized by a 2D Fourier transform matrix $\mathbf{F}_{2D}$ (as per property 3). In this assignment all practical examples will involve 2D signals (i.e. images) and therefore 2D circular shifts. The $2D$ subscript, however, will be omitted herein. For example, if you wanted to apply a 2D FFT matrix transform $\mathbf{F}$ to a vectorized 2D image patch $\mathbf{a} = \text{vec}(\mathbf{A})$ you would do this in MATLAB through the line,

```
>> Af = fft2(A);
```

where $\mathbf{A}$ is the 2D image patch, and $\hat{\mathbf{A}}$ (which is expressed as $\mathbf{Af}$ in MATLAB) is the unvectorized 2D FFT of $\mathbf{A}$ such that $\hat{\mathbf{a}} = \text{vec}(\hat{\mathbf{A}}) = \mathbf{F}\mathbf{a}$.

Now run the script file `example_circshift.m`, and observe a visual depiction of the extraction of circular-shifted sub-images from the Lena image (Figure 1 MATLAB) as well as the desired output response from our linear discriminant (Figure 2 MATLAB). Please proceed to answer the following questions:-

- a) Add your solution to Equation 1 in the `example_circshift.m` code. Visualize the resultant linear discriminant weight vector $\mathbf{g}$ using the MATLAB function `imagesc` for the penalty value λ that worked best in question 1(b) (remember to use the `reshape` command to convert $\mathbf{g}$ back into a 2D array). Apply the filter to the entire Lena image using the `imfilter` function. Visualize the response using `imagesc`. Include the visualization in your written response document. Is the circular shifted response a suitable approximation to the one obtained in question 1(b)? **Place the answer and figures in your written response. (5 points)**
- b) Using the properties described above formulate a way to learn $\mathbf{g}$ without having to generate circular shifts explicitly. Specifically we would like you to take advantage of properties 1 and 3 to simplify the solution to Equation 1 given that we assume that $\mathbf{X}$ is circulant Toeplitz such that,

$$\hat{\mathbf{g}} = (\hat{\mathbf{x}} \circ \hat{\mathbf{y}}) \circ^{-1} (\hat{\mathbf{s}} + \lambda) \quad (2)$$

where $\hat{\mathbf{s}} = \hat{\mathbf{x}} \circ \text{conj}\{\hat{\mathbf{x}}\}$ is commonly referred to as the “spectrum” of $\mathbf{x}$. We define $\circ^{-1}$ and $\circ$ as the inverse and regular Hadamard product operators (equivalent to `.*` and `./` operations in MATLAB). Modify the solution in `example_circshift.m` to apply this new approach. Apply the weight vector $\mathbf{g}$ to the entire Lena image using the `imfilter` function. Visualize the responses using `imagesc` for a value of $\lambda = 1$. Include these visualizations in your written response document. **Place the answer, full working and figures in your written response. (15 points)**

- c) Visualize the weight vector $\mathbf{h} = \mathcal{F}^{-1}\{\text{conj}(\hat{\mathbf{g}})\}$ where $\hat{\mathbf{g}} = \mathcal{F}\{\mathbf{g}\}$. Based on the property that $\mathbf{x} * \mathbf{h} = \mathcal{F}\{\mathbf{x}\} \circ \mathcal{F}\{\mathbf{h}\}$ - where $*$ is the convolutional operator and $\circ$ is the Hadamard product operator - demonstrate how Equation 1 can be rewritten using a convolutional operator between $\mathbf{x}$ and $\mathbf{h}$. **Place the answer and full working in your written response. (5 points)**
- d) One can re-frame the objective in Equation 1 to be a least-squares kernel discriminant

$$\arg \min_{\boldsymbol{\alpha}} \frac{1}{2} \|\mathbf{y} - \mathbf{K}\boldsymbol{\alpha}\|_2^2 + \frac{\lambda}{2} \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} \quad (3)$$

where for the linear case $\mathbf{g} = \mathbf{X}\boldsymbol{\alpha}$ and the kernel matrix $\mathbf{K} = \mathbf{X}^T \mathbf{X}$. Using property 3 and 4, demonstrate why if $\mathbf{X}$ is circulant Toeplitz that $\mathbf{K}$ is always diagonalizable by the Fourier transform

$$\mathbf{K} = \frac{1}{D} \mathbf{F} \text{diag}(\hat{\mathbf{k}}) \mathbf{F}^T.$$

Can you also comment on the relationship of $\hat{\mathbf{k}}$ to the spectrum $\hat{\mathbf{s}}$ of $\mathbf{x}$? **Place the answer and full working in your written response. (5 points)**

- e) For the multi-channel case, as was discussed in class, let's consider a color image with 3 channels $[\mathbf{x}^{(r)}, \mathbf{x}^{(g)}, \mathbf{x}^{(b)}]$ for red, green and blue respectively. Given that we form a circulant shifted matrix $\mathbf{X}^{(c)}$ for each channel $c \in \{r, g, b\}$ demonstrate why the concatenation of these circulant shifted matrices

$$\mathbf{X}^{(rgb)} = \begin{bmatrix} \mathbf{X}^{(r)} \\ \mathbf{X}^{(g)} \\ \mathbf{X}^{(b)} \end{bmatrix}$$

is no longer diagonalized by the Fourier transform matrix. Further, demonstrate why the kernel matrix

$$\mathbf{K}^{(rgb)} = [\mathbf{X}^{(rgb)}]^T [\mathbf{X}^{(rgb)}]$$

is still diagonalized by the Fourier transform and comment on how one can estimate $\hat{\mathbf{k}}^{(rgb)}$ efficiently. **Place the answer and full working in your written response. (5 points)**

- f) Now run the script file `multichannel_circshift.m`, and observe a visual depiction of the extraction of circular-shifted multi-channel (i.e. 3 channels - red, green, and blue) sub-images from the Lena

image (Figure 1 MATLAB) as well as the desired output response from our linear discriminant (Figure 2 MATLAB). The code also depicts the solution to this system when being solved spatially such that,

$$\arg \min_{\mathbf{g}^{(rgb)}} \frac{1}{2} \|\mathbf{y} - [\mathbf{X}^{(rgb)}]^T \mathbf{g}^{(rgb)}\|_2^2 + \frac{\lambda}{2} \|\mathbf{g}^{(rgb)}\|_2^2 . \quad (4)$$

From your solution to the previous question formulate an efficient solution to the multi-channel correlation filter such that,

$$\hat{\boldsymbol{\alpha}} = \hat{\mathbf{y}} \circ^{-1} (\hat{\mathbf{k}}^{(rgb)} + \lambda) \quad (5)$$

and

$$\hat{\mathbf{g}}^{(c)} = \hat{\mathbf{x}}^{(c)} \circ \hat{\boldsymbol{\alpha}} \text{ for } c \in \{r, g, b\} . \quad (6)$$

Can you describe why this implementation would be more efficient than the naive spatial version in Equation 4? **Place all answers and derivations in your written response, also include the implementation of your new solution at the end of the file `multichannel_circshift.m`. Visualize the result and include in your written response. (15 points)**